

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1 Identifikasi Permasalahan

Seiring perkembangan teknologi saat ini, sangat banyak aplikasi atau sistem yang dibuat dengan tujuan untuk mempermudah aktifitas manusia . Misalnya pada musik, manusia membutuhkan suatu alat atau sistem untuk melakukan penyetelan nada pada suatu alat musik, Nada merupakan bunyi yang teratur dan mempunyai frekuensi yang tertentu. Pengolahan pada frekuensi ini, sering disebut dengan istilah pengolahan sinyal digital (*digital signal processing*).

Pada pengolahan sinyal digital (*digital signal processing*) terdapat beberapa jenis algoritma yang dapat digunakan, diantaranya Fast Fourier Transform, Gortzel, Hamming Window, transformasi wavelet packet dan lain – lain. Salah satu algoritma yang sering digunakan pada pengolahan sinyal digital adalah Algoritma *Fast Fourier Transform* (FFT), FFT diklaim dapat bekerja dengan baik sehingga menghasilkan akurasi dengan cepat dan efisien.

Sehingga, untuk membuktikan *Fast Fourier Transform* (FFT) dapat bekerja dengan baik dan efisien, Salah satu cara untuk menyelesaikan masalah tersebut adalah dengan menghitung tingkat akurasi dari algoritma *Fast Fourier Transform* (FFT) pada tuning gitar. Dengan penelitian ini, diharapkan dapat membuktikan dan menghasilkan aplikasi tuner gitar yang efisien.

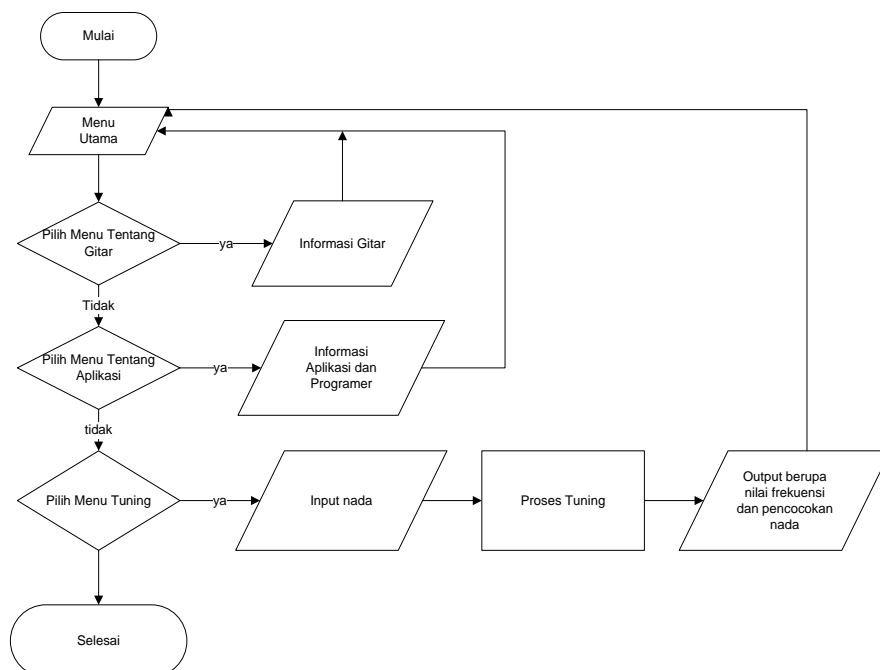
4.2 Analisis Sistem

Analisis sistem merupakan bagian penelitian yang menganalisis sistem yang ada untuk merancang sistem baru atau memperbaharui sistem yang ada. Bagian ini merupakan bagian yang penting dikarenakan hasil dari sistem yang akan dibuat tergantung dari analisis yang dilakukan.

4.2.1 Pemahaman Kerja Sistem yang ada

4.2.1.1 Alur Sistem

Alur sistem merupakan hasil analisis perancangan tahapan kerja sistem yang akan dibangun. Alur ini dimulai dari user memasukkan *input*-an data sampai dengan menghasilkan keluaran *output*. Dalam sistem ini, *input* berupa nada yang dihasilkan oleh senar gitar dan *output*-nya berupa nilai frekuensi dan pencocokan nada (dapat dilihat pada bab 3, bagian 3.5). Secara garis besar tahapan perancangan sistem yang akan dibangun dapat dilihat pada Gambar 4.1.



Gambar 4.1 Diagram Alir Sistem

Berdasarkan diagram alir pada Gambar 4.1, terdapat beberapa pilihan menu yang merupakan bagian dari sistem yang akan dibangun, menu-menu tersebut adalah sebagai berikut:

1. Menu Tentang Gitar

Dalam menu ini, ditampilkan informasi mengenai gitar. Informasi ini berupa pengenalan tentang gitar, sejarah gitar dan jenis-jenis gitar.

2. Menu Tentang Aplikasi

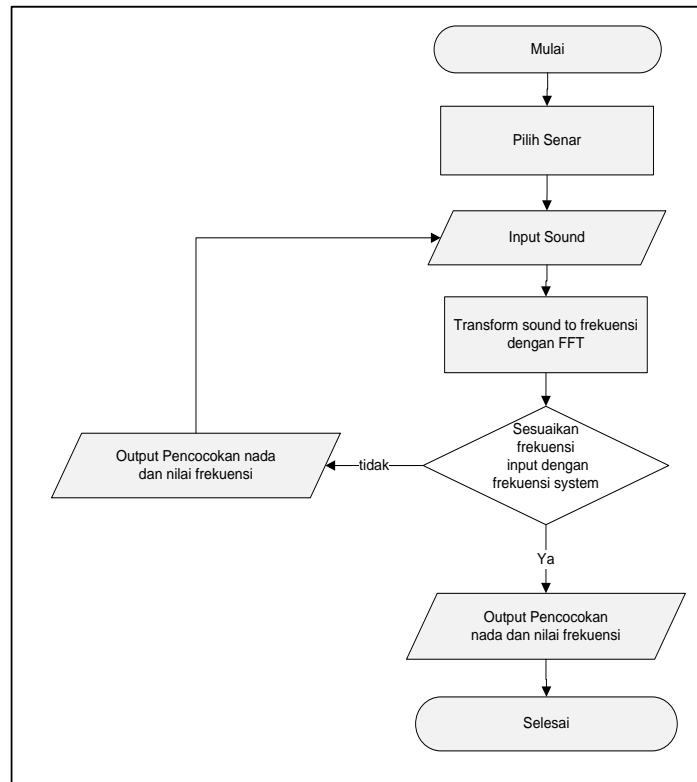
Seperti menu Tentang Tuner Gitar, menu ini menampilkan informasi pula, hanya saja informasi yang ditampilkan adalah informasi tentang aplikasi ini.

3. Menu Tuning

Menu ini merupakan bagian inti dari aplikasi. Dalam menu ini, user perlu mengaktifkan button sebagai on off dari aplikasi tuner. Kemudian proses input akan berjalan runtime. Input tersebut akan diproses, setelah diproses. Maka akan tampil *output* berupa nilai frekuensi dan pencocokan nada (lihat pada bab 3, bagian 3.5).

4.2.2 Alur Algoritma Fast Fourier Transform

Dalam sistem yang akan dibangun ini, terdapat suatu algoritma yang digunakan yaitu algoritma Fast Fourier Transform. Algoritma ini akan digunakan untuk melakukan perhitungan Discrete Fourier Transform (DFT), dimana DFT sendiri adalah proses perubahan (transform) dari kawasan waktu (diskrit) menjadi frekuensi. Alur dari tahapan ini diperlihatkan pada Gambar 4.2 sebagai berikut.



Gambar 4.2 diagram alir pada citra uji

Dari gambar tersebut dapat kita lihat, proses pertama adalah memilih senar. Kemudian dilanjutkan dengan memetik senar tersebut dan didapatkanlah masukkan untuk sistem. Masukkan diproses dengan menggunakan algoritma Fast Fourier Transform untuk mendapatkan nilai frekuensi. Nilai frekuensi inilah yang nantinya akan disesuaikan dengan sistem dan mendapatkan hasil berupa nilai frekuensi dan kecocokan nada senar.

4.2.2 Analisis Fungsional

Analisis fungsional berarti melakukan analisis fungsi-fungsi pada sistem yang akan dibangun. Fungsi-fungsi pada sistem yang dimaksud adalah fitur-fitur apa saja yang ada didalam sistem. Berikut ini merupakan beberapa fitur yang ada pada sistem yang akan dibangun.

1. Mampu melakukan penyetelan nada (*Tuning*) dengan menggunakan algoritma Fast Fourier Transform (FFT).

2. Memberikan informasi mengenai tuner gitar yaitu berupa pengenalan gitar, sejarah dan jenis gitar.

4.2.3 Analisis Non-Fungsional

Analisis non-fungsional berarti melakukan analisis yang berhubungan dengan sistem diluar fitur-fitur yang akan dibangun dalam sistem berupa perangkat lunak dan perangkat keras yang mendukung dalam pembangunan sistem. Berikut ini merupakan beberapa perangkat keras dan perangkat lunak yang mendukung dalam pembangunan sistem.

1. Kebutuhan Perangkat Lunak (*software*)

Perangkat lunak yang mendukung aplikasi dalam penelitian ini adalah berupa Sistem Operasi *Windows Seven* (7) 32 bit, Matlab 2008, dan *Microsoft Office Visio 2007* untuk pembuatan diagram alir sistem, *data flow diagram* dan merancang form sistem.

2. Kebutuhan Perangkat Keras (*hardware*)

Sedangkan perangkat keras yang mendukung dalam penelitian ini adalah 1 unit Laptop Acer tipe Aspire dengan spesifikasi monitor VGA (1366 x 768) dan *processor* Intel Core i3, RAM 4 GB, Harddisk 500 GB dan printer *Canon IP2770*.

4.3 Perancangan Sistem

Perancangan sistem merupakan tahap selanjutnya dalam membangun sebuah sistem setelah melakukan identifikasi masalah dan analisis sistem. Tujuan dalam melakukan perancangan sistem ini adalah untuk memberikan gambaran yang jelas dan lengkap tentang rancang bangun dan implementasi dari sistem yang akan dibuat. Perancangan yang dilakukan dalam sistem ini

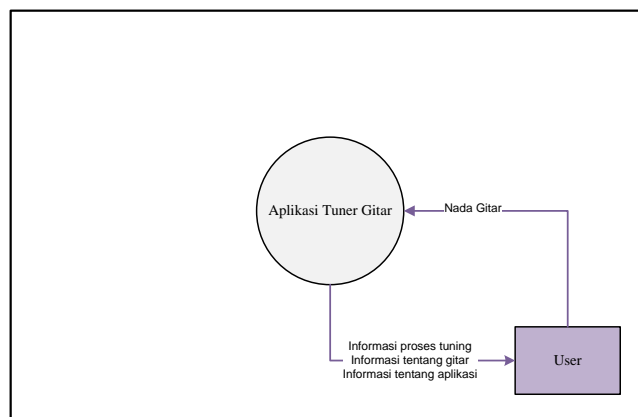
adalah perancangan *Data Flow Diagram* (DFD) dan perancangan antarmuka sistem.

4.3.1 Perancangan *Data Flow Diagram* (DFD)

Perancangan *Data Flow Diagram* (DFD) ini ditujukan untuk memberikan gambaran secara umum tentang aplikasi yang akan dibangun dalam penelitian ini. Perancangan ini dibuat dalam tiga bagian level, yaitu diagram konteks, diagram level 0 dan diagram level 1.

1. Diagram Konteks atau Diagram Level 0

Diagram konteks merupakan diagram tertinggi yang ada di *Data Flow Diagram* (DFD). Dalam diagram ini menggambarkan hubungan sistem dengan lingkungan disekitar sistem. Tujuan dari pembuatan diagram konteks ini adalah memberikan pandangan tentang aplikasi yang akan dibangun secara umum. Berbeda dengan diagram pada level lainnya, pada diagram konteks hanya memiliki satu proses saja. Proses tersebut merupakan proses inti dari sistem yang akan dibuat. Dalam diagram konteks, Diagram konteks dari aplikasi atau sistem yang akan dibangun diperlihatkan pada Gambar 4.5 berikut ini.



Gambar 4.3 Digram Konteks Aplikasi Tuner Gitar

Pada Gambar 4.3 diperlihatkan diagram konteks dari sistem yang akan dibuat. Dalam diagram tersebut terdapat hanya satu entitas yang merupakan pihak atau orang yang berinteraksi terhadap sistem yang akan dibangun. Entitas pada sistem ini yaitu user, entitas tersebut memiliki hak akses didalam sistem. Adapun hak akses dari entitas tersebut dapat dijelaskan pada aliran data masukan (*input*) dan data keluaran (*output*) pada Tabel 4.1 berikut ini.

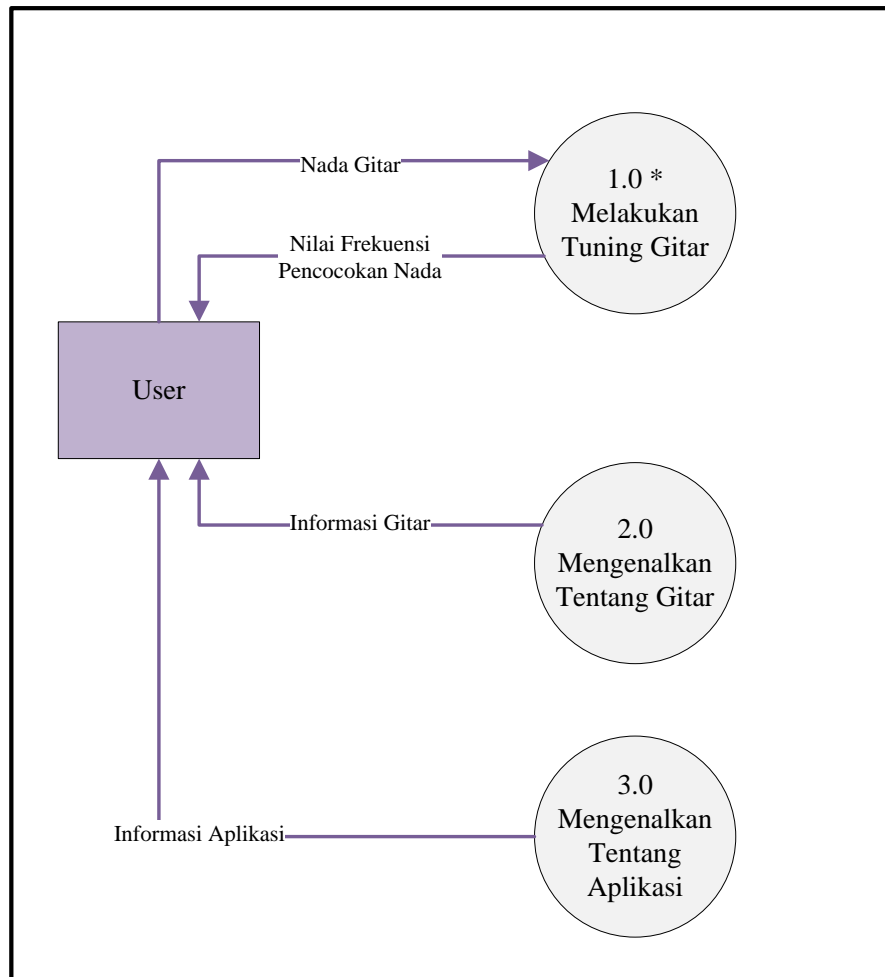
Tabel 4.1 Aliran data entitas User

Entitas	Aliran data	
	Masukan (<i>input</i>)	Keluaran (<i>output</i>)
User	Nada Gitar	Informasi Proses Tuning (Nilai Frekuensi dan Informasi Pencocokan Nada)
	-	Informasi Gitar
		Informasi Aplikasi

Pada tabel 4.1 diperlihatkan aliran data sebagai masukan (*input*) dan keluaran (*output*). User dapat memasukkan nada gitar kedalam sistem dan mendapat *feed back* yaitu hasil dari tuning gitar berdasarkan nada yang dimasukkan. Selain user mengetahui nilai frekuensi dan informasi pencocokan nada, user dapat mengetahui juga mengenai informasi gitar dan informasi mengenai aplikasi.

2. Diagram Level 1

Seperti yang telah dijelaskan pada bab tinjauan pustaka, diagram level 1 merupakan pemecahan dari diagram sebelumnya yaitu diagram, konteks. Pada diagram level 1 ini terdapat 3 proses yang menggambarkan aplikasi tuner gitar yang akan dibangun. Diagram level 1 dari aplikasi tuner gitar ini dapat dilihat pada Gambar 4.4.



Gambar 4.4 Diagram Level 1 Aplikasi Tuner Gitar

Gambar 4.4 memperlihatkan diagram level 1 dari aplikasi tuner gitar yang akan dibangun. Dalam diagram tersebut, terdapat 3 proses yang menggambarkan aplikasi. Adapun ketiga proses tersebut akan dijelaskan sebagai berikut.

a. Proses 1.0 * Melakukan Tuning Gitar

Proses 1.0 * dilakukan oleh entitas user, user perlu memasukkan nada gitar yang dijadikan signal uji. Kemudian akan dilakukan penyetelan nada,. *Feed back*-nya user akan memperoleh informasi hasil tuning gitar yang telah dilakukan. Dalam proses 1.0 ini dalam gambar 4.3 terlihat ada

tanda bintang (*) setelah nomor dari proses tersebut. tanda bintang (*) tersebut dimaksudkan untuk menandakan bahwa proses tersebut memiliki proses yang lebih rinci yang dijelaskan pada diagram rinci. Itu berarti proses mengelola tuning ini memiliki proses yang lebih rinci yang akan dijelaskan pada diagram level 2 pada Gambar 4.4.

b. Proses 2.0 Mengenalkan Tentang Gitar

Dalam proses 2.0 ini, sistem dapat mengenalkan sedikit pengetahuan tentang Gitar yang dijadikan sebagai studi kasus pada penelitian ini. entitas user mendapatkan informasi tentang Batik Besurek tersebut.

c. Proses 3.0 Mengenalkan Tentang Aplikasi

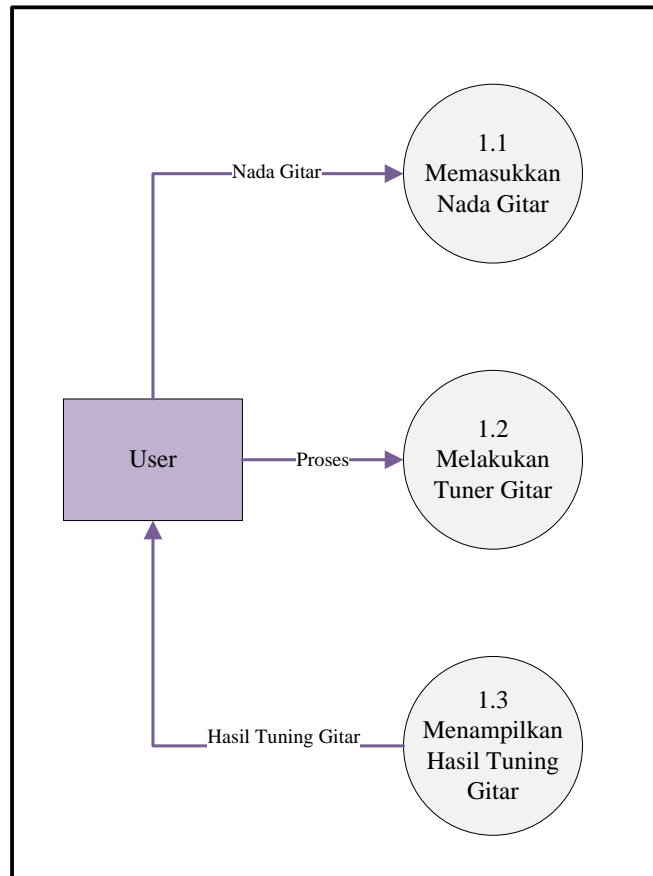
Dalam proses 3.0 ini, sistem dapat mengenalkan bagaimana menggunakan aplikasi ini. Sehingga diharapkan user baru yang menggunakan sistem, dapat dengan mudah beradaptasi.

3. Diagram Level 2

Diagram ini merupakan diagram rinci yang akan menjelaskan proses-proses lebih jelas dari diagram level sebelumnya yaitu diagram level 1. Diagram level 2 pada sistem yang akan dibangun ini terdiri atas satu diagram yaitu diagram level 2 proses 1 yaitu melakukan tuning gitar.

a. Diagram Level 2 Proses 1 Melakukan Tuning Gitar

Diagram Level 2 Proses 1 ini yaitu proses melakukan tuning gitar. Proses ini merupakan rincian dari proses 1 pada diagram level 1 yang diperlihatkan pada Gambar 4.4. Berikut ini adalah Gambar 4.5 yang merupakan diagram level 2 proses 1 yaitu melakukan pencarian citra.

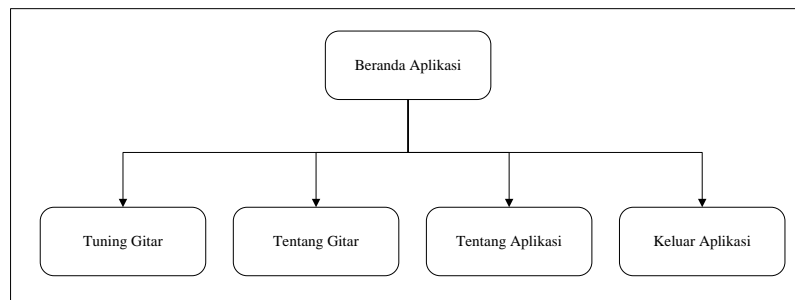


Gambar 4.5 Diagram Level 2 Proses 1 Aplikasi Tuner Gitar

Gambar 4.5 memperlihatkan diagram level 2 pada proses 1 yaitu melakukan tuning gitar. Proses ini dibagi menjadi 3 sub proses yaitu proses 1.1 memasukkan nada gitar, yang kedua proses 1.2 melakukan tuner gitar dan yang ketiga 1.3 menampilkan hasil tuning gitar. Proses 1.1 adalah proses dimana user memasukkan nada gitar sebagai signal uji kedalam sistem yang akan dibangun ini. Proses kedua yaitu 1.2, user akan menekan tombol yang disediakan untuk melakukan proses tuning oleh sistem. Proses ini akan menggunakan data masukkan yang telah diinput secara runtime pada proses pertama. Proses yang ketiga yaitu 1.3 menampilkan hasil tuning gitar. hasil tuning gitar tersebut dapat berupa nilai frekuensi dan pencocokan nada

4.3.2 Perancangan Antarmuka

Antarmuka (*interface*) merupakan suatu media interaksi antara pengguna dengan aplikasi yang akan dibangun. Dengan demikian, antarmuka merupakan bagian penting dalam pembuatan sebuah sistem atau aplikasi. Sebelum membangun antarmuka yang nyata diperlukan perancangan agar menghasilkan antarmuka yang sesuai dengan keinginan dan kebutuhan sistem itu sendiri. Untuk itu perlunya dibuat struktur antarmuka itu sendiri. Berikut ini Gambar 4.6 yang merupakan struktur dari antarmuka aplikasi tuner gitar.



Gambar 4.6 Struktur Antarmuka Aplikasi

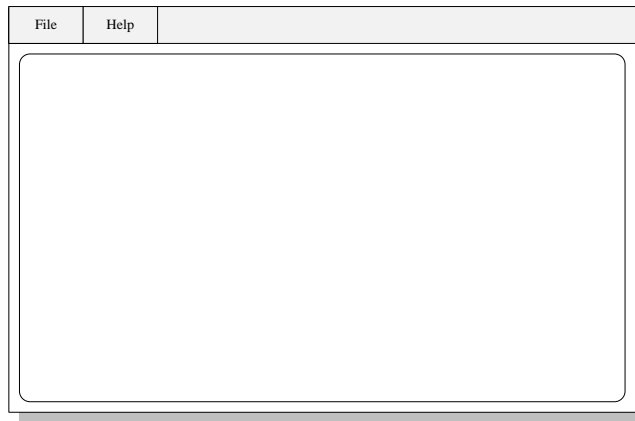
Aplikasi dimulai dari beranda aplikasi seperti yang ditunjukkan Gambar 4.6. Kemudian ada empat cabang antarmuka yaitu tuning gitar, antarmuka tentang gitar, tentang aplikasi dan keluar aplikasi.

Setelah struktur antarmuka aplikasi dibuat, maka berikutnya dilakukan perancangan antarmuka sesuai dengan struktur antarmuka yang telah dibuat. Berikut ini merupakan perancangan antarmuka dari aplikasi tuner gitar yang dibuat sesuai dengan kebutuhan sistem.

1. Beranda Aplikasi

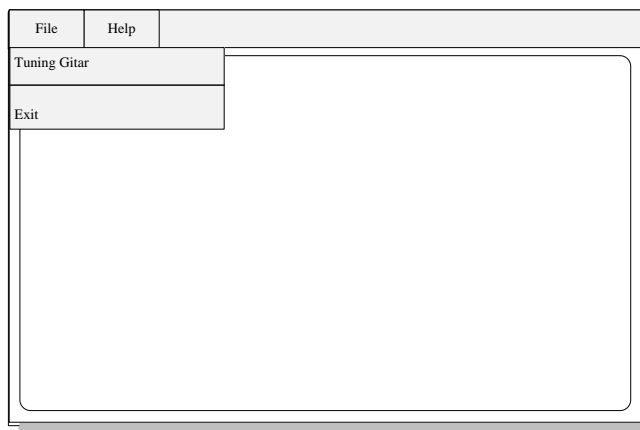
Beranda aplikasi merupakan tampilan awal dari aplikasi ketika aplikasi tuner gitar ini dijalankan. Gambar 4.7 menunjukkan perancangan

antarmuka beranda aplikasi dalam penelitian ini. Didalam beranda aplikasi, terdapat dua menu yaitu menu file dan help yang terletak dibagian atas kiri beranda ini.



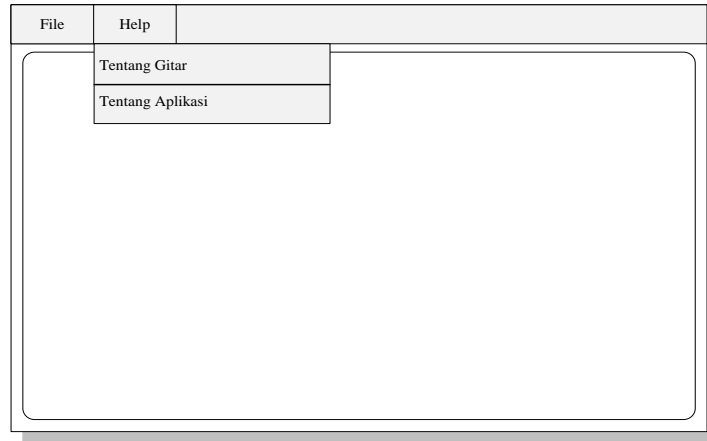
Gambar 4.7 Menu Utama Aplikasi

Kedua menu yang diperlihatkan pada Gambar 4.7, masing-masing menunya memiliki sub menu yang berbeda. Menu file memiliki dua sub menu yaitu tuning gitar, dan submenu Exit seperti yang ditunjukkan pada Gambar 4.8.



Gambar 4.8 Sub Menu File

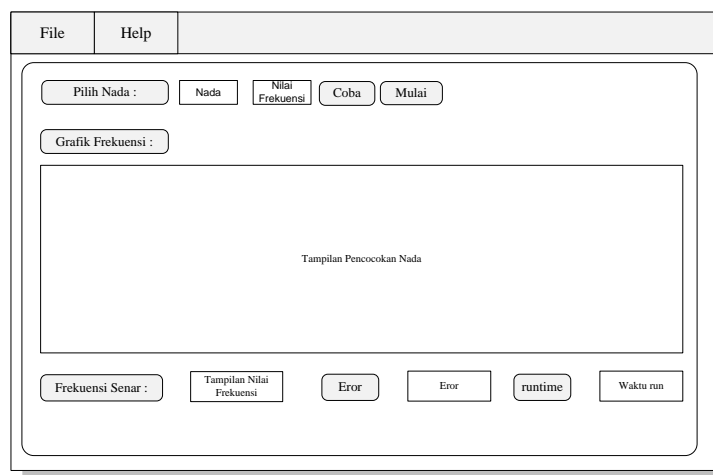
Menu berikutnya yaitu menu Help yang berisi submenu Tentang Gitar dan submenu Tentang Aplikasi. Tampilan isi dari submenu Help ditunjukkan pada Gambar 4.9 berikut ini.



Gambar 4.9 Sub Menu Help

2. Antarmuka Tuning Gitar

Antarmuka aplikasi tuning gitar ini merupakan tampilan setelah memilih submenu tuning gitar yang ada di menu File. Gambar 4.10 menunjukkan tampilan tuning tersebut. Didalam antarmuka ini terdapat button untuk mengaktifkan dan menonaktifkan input nada gitar secara runtime, tampilan untuk menampilkan nilai frekuensi dan tampilan pencocokan nada.

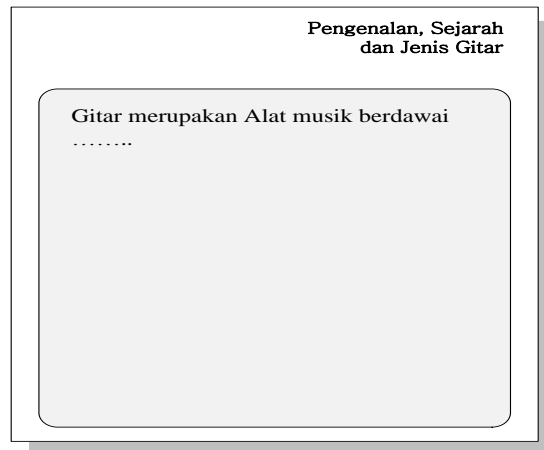


Gambar 4.10 Form Tuning Gitar

3. Antarmuka submenu Tentang Gitar

Tampilan submenu Tentang Gitar ini berisikan penjelasan ataupun

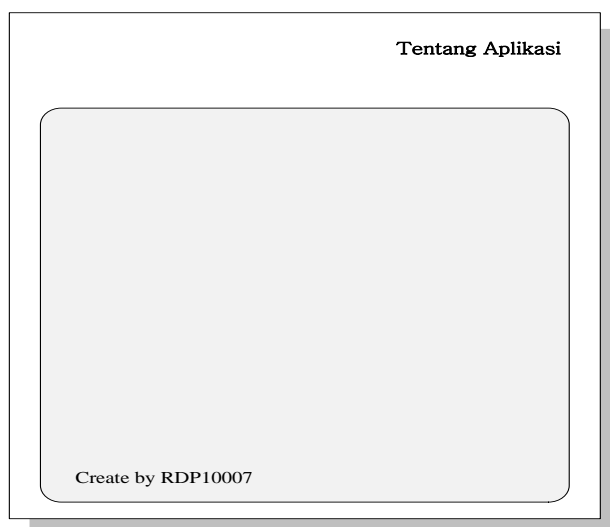
uraian singkat mengenai Gitar mulai dari pengenalan Gitar itu sendiri, sejarahnya dan juga jenis-jenis yang ada pada Gitar itu sendiri. Tampilan submenu ini diperlihatkan pada Gambar 4.11 berikut ini.



Gambar 4.11 Form Tentang Gitar

4. Antarmuka submenu Tentang Aplikasi

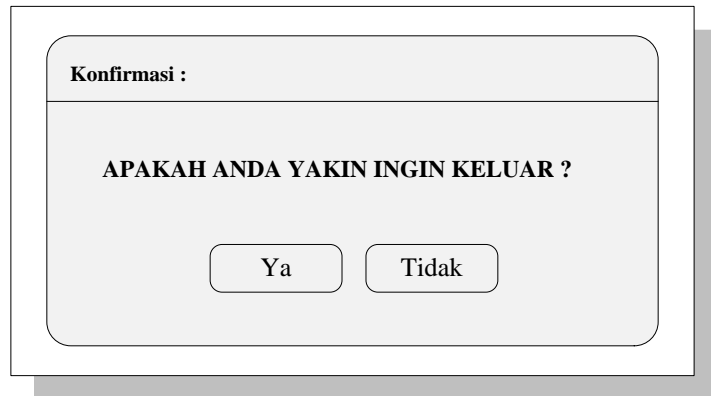
Tampilan submenu tentang aplikasi ini berisikan penjelasan ataupun uraian singkat tentang aplikasi. Tampilan antarmuka submenu tentang aplikasi dapat dilihat pada Gambar 4.12.



Gambar 4.12 Form Tentang Aplikasi

5. Antarmuka submenu *Exit*

Tampilan submenu *Exit* aplikasi ini menunjukkan proses keluar dari aplikasi. Jika pengguna ingin keluar aplikasi maka akan keluar kotak dialog untuk keluar aplikasi. Tampilan keluar aplikasi dapat dilihat pada Gambar 4.13.



The image shows a standard Windows-style confirmation dialog box. It has a light gray background and a thin black border. At the top, there is a header bar with the text 'Konfirmasi :'. Below this, the main area contains the question 'APAKAH ANDA YAKIN INGIN KELUAR ?' in all caps. At the bottom, there are two buttons: 'Ya' on the left and 'Tidak' on the right, both with rounded corners and a light gray fill.

Gambar 4.13 Form Keluar Aplikasi

BAB V

HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai hasil dan pembahasan aplikasi yang telah dibuat berdasarkan analisis dan perancangan sistem yang telah dijelaskan pada bab sebelumnya. Penjelasan pada bab ini terdiri dari implementasi antar muka sistem, pengujian *white box* dan *black box*.

5.1 Implementasi Antar Muka

Hasil dari analisis dan perancangan sistem yang telah dilakukan sebelumnya akan mempengaruhi hasil dari implementasi antar muka sistem. Pada tahapan implementasi antar muka ini, sistem akan diimplementasikan menggunakan bahasa pemrograman MATLAB versi **7.7.0 (R2008b)**. Berikut ini merupakan file-file yang digunakan untuk mengimplementasikan antar muka sistem.

Tabel 5.1 Daftar m.file dan *Figure* Aplikasi

No	Nama File	Deskripsi
1.	a) TentangGitar.m	File untuk membuat halaman tentang gitar
	b) TentangGitar.fig	Desain tampilan dari halaman tentang gitar
2.	a) TentangAplikasi.m	File untuk membuat halaman mengenai aplikasi
	b) TentangAplikasifig	Desain tampilan dari halaman tentang gitar
3.	a) Home.m	File untuk membuat halaman utama pada aplikasi
	b) Home.fig	Desain tampilan dari halaman utama pada aplikasi
4.	a) GuitarTuner.m	File untuk membuat halaman tuning gitar pada aplikasi
	b) GuitarTuner.fig	Desain tampilan dari halaman tuning gitar
5.	a)ButtonIcon.mat	Desain tombol

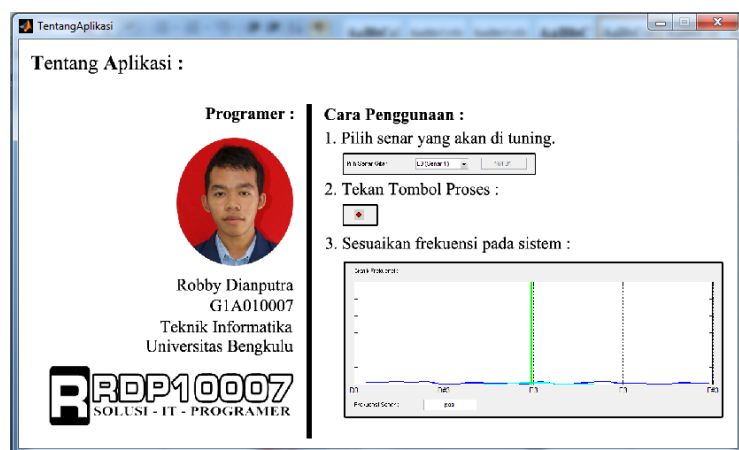
Tabel 5.1 merupakan tabel yang mendeskripsikan nama-nama dan kegunaan dari setiap .m dan *figure* dari sistem. Terlihat bahwa terdapat 8 .m, 1 file .mat dan 8 file *figure* yang digunakan dalam pembuatan aplikasi tuning gitar ini. File .m adalah file yang berekstensi .m yang merupakan file yang digunakan untuk

mengimplementasikan perancangan dalam bentuk barisan kode untuk mendukung berjalannya file *figure*. Sedangkan file *figure* merupakan file yang digunakan untuk mengimplementasikan hasil perancangan dalam bentuk antarmuka aplikasi (*layout*). Berikut ini adalah file *figure* yang digunakan sebagai antarmuka aplikasi terhadap pengguna.



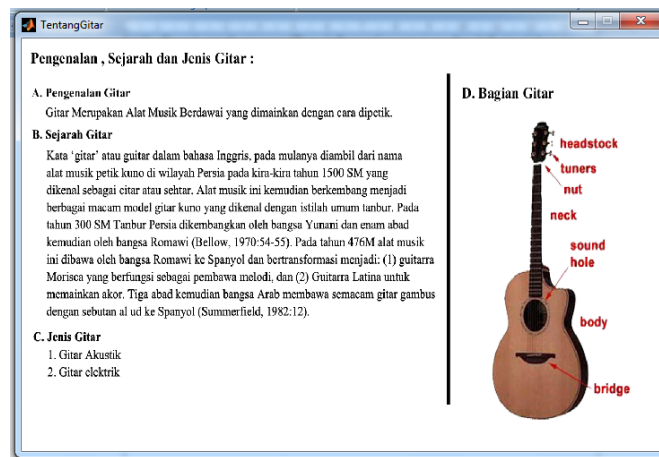
Gambar 5.1 Home.fig

Gambar 5.1 menunjukkan tampilan dari halaman utama aplikasi. Halaman utama merupakan halaman pertama yang akan keluar saat sistem dijalankan. Pada halaman utama ini terdapat judul dari aplikasi yang telah dibuat ini yaitu “Aplikasi Tuning Gitar”. Halaman utama ini memiliki beberapa gambar yang merupakan gambar dari gitar itu sendiri dan tulisan.



Gambar 5.2 TentangAplikasi.fig

Gambar 5.2 merupakan tampilan dari file form TentangAplikasi.fig yang menggambarkan salah satu menu yang ada didalam aplikasi yang dibuat ini. Menu ini berfungsi untuk menampilkan informasi programer dan informasi bagaimana menggunakan aplikasi ini.



Gambar 5.3 TentangGitar.fig

Dalam aplikasi ini, selain menu tentang gitar, terdapat menu lain yang berfungsi untuk menampilkan informasi tentang gitar. Gambar 5.3 menunjukkan tampilan dari menu tentang gitar tersebut. Pada halaman ini terdapat beberapa informasi seperti pengenalan , sejarah dan jenis gitar serta bagian-bagian gitar.

Aplikasi ini mengambil studi kasus yaitu tuning gitar. Dengan demikian, didalam aplikasi ini dicantumkan sebuah form yang mampu melakukan tuning seperti pada gambar 5.4



Gambar 5.4 GuitarTuner.fig

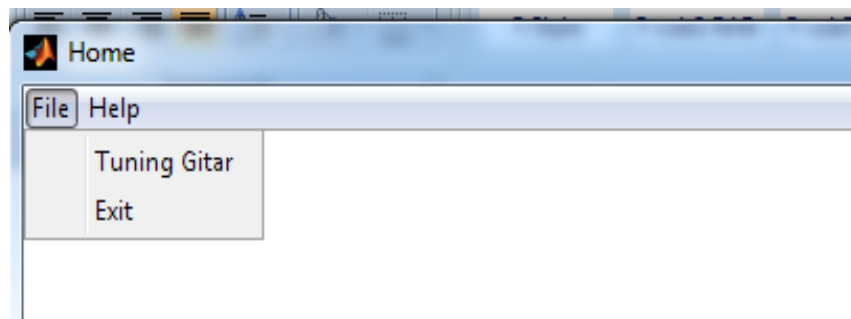
5.2 Pengujian Sistem

5.2.1 Pengujian White Box

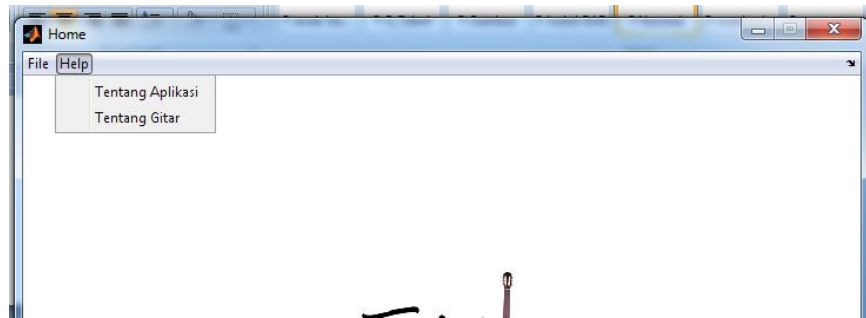
Pengujian white box dilakukan dengan menguji kode-kode program yang dibuat pada aplikasi. Pengujian dilakukan dengan mengecek semua kode pada program telah dieksekusi paling tidak satu kali. Pengujian ini dilakukan pada proses pengembangan sistem yakni pengujian kode program (*coding*). Adapun tampilan dan potongan *source code* untuk setiap menu pada aplikasi ini adalah sebagai berikut:

1) Beranda Aplikasi

Seperti yang telah dijelaskan sebelumnya, beranda aplikasi merupakan halaman yang pertama kali diakses ketika aplikasi ini dijalankan. Pada beranda aplikasi ini terdapat dua menu yang digunakan untuk masuk ke menu yang ada didalam aplikasi ini yaitu menu File dan menu Help. Masing-masing menu memiliki sub menu yang berbeda-beda. Berikut ini Gambar 5.5 yang merupakan tampilan beranda aplikasi yang menunjukkan isi menu File, sedangkan Gambar 5.6 merupakan tampilan beranda aplikasi yang menunjukkan isi menu Help



Gambar 5.5 Isi Menu File pada Beranda Aplikasi



Gambar 5.6 Isi Menu Help pada Beranda Aplikasi

Berikut ini merupakan potongan *source code* menu File dan menu Help pada beranda aplikasi.

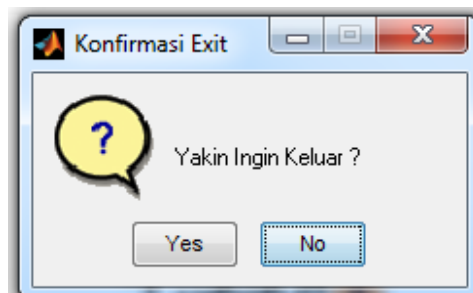
Tabel 5.2 source code menu file dan menu help

	... fungsi standar pada matlab, yang tidak boleh diedit
1	<code>function M_TuningGitar_Callback(hObject, eventdata, handles)</code>
2	<code>delete(Home);</code> <code>figure(GuitarTuner);</code>
3	<code>function M_Exit_Callback(hObject, eventdata, handles)</code>
4	<code>button = questdlg('Yakin Ingin Keluar ?', 'Konfirmasi Exit', 'Yes', 'No', 'No');</code> <code>switch button</code> <code>case 'Yes' ,</code> <code>delete(Home);</code> <code>case 'No',</code> <code>quit cancel;</code> <code>end</code>
5	<code>function M_TentangAplikasi_Callback(hObject, eventdata, handles)</code>
6	<code>delete(Home);</code> <code>figure(TentangAplikasi);</code>
7	<code>function M_TentangGitar_Callback(hObject, eventdata, handles)</code>
8	<code>delete(Home);</code> <code>figure(TentangGitar);</code>

Berdasarkan potongan *source code* table 5.2 , terdapat kode *figure*, kode *delete* dan kode *questdlg*. *Figure* digunakan untuk memanggil form lain yang terdapat didalam aplikasi ini. `Figure(TentangGitar)` digunakan untuk memanggil form `TentangGitar`. Form `TentangGitar` adalah form untuk masuk kedalam sub menu informasi mengenai gitar.

Figure (TentangAplikasi) digunakan untuk memanggil form sub menu TentangAplikasi, begitu juga dengan kode *figure* lainnya memiliki fungsi yang sama. Kode *delete* digunakan untuk menghapus *figure* yang sedang ditampilkan.

Sedangkan *questdialog* atau yang berarti *question dialog* digunakan untuk memanggil kotak dialog pertanyaan. Dalam beranda aplikasi ini, digunakan kotak dialog untuk menanyakan kepada pengguna apakah benar-benar ingin keluar dari aplikasi ini. Berikut ini adalah Gambar 5.7 yang menunjukkan kotak dialog yang digunakan.



Gambar 5.7 Kotak Dialog Keluar Beranda Aplikasi

2) Menu File

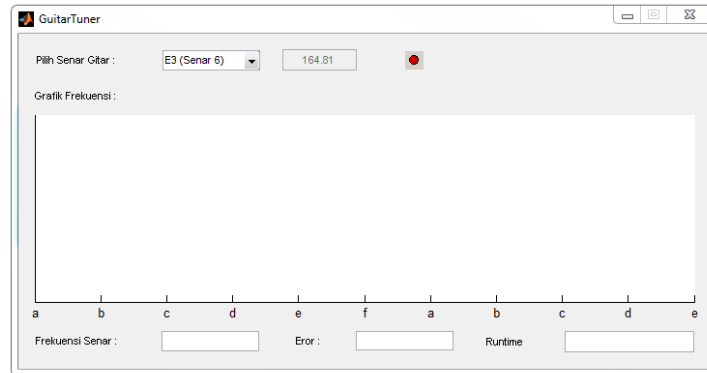
Berdasarkan Gambar 5.5, gambar tersebut menunjukkan isi dari menu File. Masing-masing sub menu tersebut akan diperlihatkan dan dijelaskan pada bagian ini.

a) Sub Menu Tuning Gitar

Kegunaan sub menu ini telah dijelaskan secara singkat pada bagian implementasi. Pada bagian ini, akan dijelaskan *source code* yang merupakan implementasi dari perancangan yang menunjang dari tampilan sub menu tuning gitar ini.

Ketika sub menu tuning gitar dipilih berdasarkan beranda aplikasi

seperti yang ditunjukkan pada Gambar 5.5, maka aplikasi akan masuk ke form *tuning gitar*. menu tuning gitar adalah fasilitas utama pada aplikasi ini, form tuning gitar ditampilkan pada gambar 5.8.



Gambar 5.8 Tuning Gitar

Form *tuning gitar* yang ditunjukkan pada Gambar 5.8 terdiri dari *edit text* ‘freq1’ dan ‘flashfreq’, *axes* “calaxes1”, *static text*, dan *pop menu* “senar” serta *push button* ‘testrekam’. Berikut ini merupakan potongan *source code* dari tuning gitar untuk masuk ke sub menu tuning gitar.

Tabel 5.3 source code sub menu tuning gitar

	... fungsi standar pada matlab, yang tidak boleh diedit
1	<pre>load ButtonIcons; set(handles.testrekam,'CData',RecDisabled); disp('GuitarTuner, by Robby Dianputra, juni 2014');</pre>
2	<pre>hback = axes('units','normalized','position',[0 0 1 1]); uistack(hback,'bottom'); [back map]=imread('TunerGitar.png'); image(back) colormap(map) set(hback,'handlevisibility','off','visible','off')</pre>
3	<pre>function freq1_CreateFcn(hObject, eventdata, handles) if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor')) set(hObject,'BackgroundColor','white'); end</pre>
4	<pre>function outstr = getNotes(tone) notes = {'C','C#','D','D#','E','F','F#','G','G#','A','A#','B'}; for n=1:length(tone), octave = floor((tone(n)-1)/12); note = mod(tone(n)-1,12)+1; outstr{n} = [notes{note},num2str(octave)]; end;</pre>

6	<pre> function tune(eventdata, handles) ... proses inisialisasi variabel fftdata = abs(fft(data,NFFT)/L); % memulai proses FFT ... proses FFT lanjutan if fmax>flat && fmax<sharp% pencocokan hold on; disp('cakep'); set(handles.flashfreq, 'String', num2str(fmax)); semilogx([1 1]*latestfit, [0 20], 'g', 'LineWidth', 3); hold off; .. </pre>
7	<pre> function testrekam_Callback(hObject, eventdata, handles) load ButtonIcons; set(handles.flashfreq, 'String', num2str('')); if get(hObject, 'UserData') set(hObject, 'UserData', 0); set(hObject, 'CData', RecDisabled) else set(hObject, 'UserData', 1); set(hObject, 'CData', RecEnabled) tune(eventdata, handles); % end; </pre>
8	<pre> function popupmenu1_Callback(hObject, eventdata, handles) tones = [41 46 51 56 60 65]; selection = get(hObject, 'Value'); if selection<7, set(handles.freq1, 'String', num2str(15.43379*exp(0.05776234 *tones(selection)), '%3.2f')); set(handles.flashfreq, 'String', num2str(15.43379*exp(0.0577 6234*tones(selection)), '%3.2f')); set(handles.freq1, 'Enable', 'off'); else set(handles.freq1, 'Enable', 'on'); end; </pre>
9	<pre> function figure1_CloseRequestFcn(hObject, eventdata, handles) delete(hObject); figure(Home); </pre>

Berdasarkan potongan *source code* tabel 5.3 , terdapat sintaks *load, get, set, disp, getNoted* dan *,fft,.* Sintaks *load* digunakan untuk meloading gambar tombol, ini bisa dilihat pada potongan *source code* 5.3 bagian 1, Sintaks *get* digunakan untuk mengambil data dari *gui* sesuai dengan ketentuan yang ditentukan. Sintaks *set* hampir sama dengan formatnya dengan sintaks *get*, hanya saja sintaks ini digunakan untuk menampilkan nilai yang diinginkan. Sintaks berikutnya yaitu *delete* dan *figure* telah dijelaskan pada penjelasan potongan *source code*

sebelumnya. Sedangkan sintaks *disp* sama saja kegunaannya dengan sintaks *set* hanya untuk menampilkan. Untuk *getNote* adalah fungsi untuk mengambil note nada-nada yang telah dibuat sebelumnya pada 4. Dan yang terakhir adalah sintak *fft*, yaitu sintaks untuk mengubah inputan menjadi nilai frekuensi.

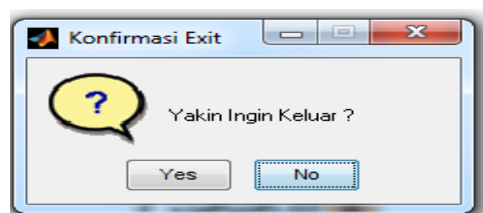
b) Sub Menu Exit

Sub menu exit merupakan bagian untuk keluar dari aplikasi, ini telah dijelaskan sebelumnya. Berikut ini potongan source code dari sub menu exit

Tabel 5.4 source code sub menu exit

	. . .
1	<code>function Untitled_4_Callback(hObject, eventdata, handles)</code>
2	<code>button = questdlg('Yakin Ingin Keluar ?', 'Konfirmasi Exit', 'Yes', 'No', 'No'); switch button case 'Yes' , delete(Home); case 'No', quit cancel; end</code>

Pada potongan source code 5.4 , terdapat sintaks *questdlg* , *switch case* , *delete* dan *quit*. Sebelumnya sintak *questdlg* dan *delete* telah dijelaskan pada beranda aplikasi gambar 5.5. sintaks *switch case* digunakan untuk pemilihan kondisi seperti *if else*, sedangkan *quit* digunakan untuk keluar dari aplikasi. Gambar 5.9 merupakan tampilan yang dihasilkan pada sintak ini.



Gambar 5.9 Tampilan Form exit

3) Menu Help

a) Sub Menu Tentang Aplikasi

Sub menu tentang Aplikasi merupakan bagian untuk menjelaskan secara singkat tentang programer dan aplikasi seperti yang telah dijelaskan dan diperlihatkan tampilannya pada Gambar 5.5. Berikut ini merupakan potongan *source code* dari sub menu Tentang Aplikasi.

Tabel 5.5 source code sub menu tentang aplikasi

	. . .
1	hback = axes('units','normalized','position',[0 0 1 1]);
2	uistack(hback,'bottom');
3	[back map]=imread('TentangAplikasi.png');
4	image(back)
6	colormap(map)
7	set(hback,'handlevisibility','off','visible','off')
8	function figure1_CloseRequestFcn(hObject, eventdata, handles)
9	delete(hObject); figure(Home);
	. . .

Pada potongan *source code* tabel 5.5, terdapat sintaks *axes*, *imread*, *set*, *delete* dan *figure*. Sintaks *axes*, *imread*, *set* telah dijelaskan pada beberapa bagian sebelum ini. Hanya saja sintaks *axes* pada potongan *source code* diatas juga diatur posisi dari *axes* tersebut diletakkan pada tampilan. Sintaks *uistack* digunakan untuk mengatur *axes* ditampilkan dibagian paling belakang sebagai latar belakang. Sedangkan *delete* dan *figure* telah dijelaskan pada gambar 5.5

b) Sub Menu Tentang Gitar

Sub menu tentang Aplikasi merupakan bagian untuk menjelaskan secara singkat tentang programer dan aplikasi seperti yang telah dijelaskan dan diperlihatkan tampilannya pada Gambar 5.5. Berikut ini

merupakan potongan *source code* dari sub menu Tentang Aplikasi.

Tabel 5.6 source code sub menu tetang gitar

	. . .
1	hback = axes('units','normalized','position',[0 0 1 1]);
2	uistack(hback,'bottom');
3	[back map]=imread('TentangGitar.png');
4	image(back)
6	colormap(map)
7	set(hback,'handlevisibility','off','visible','off')
8	function figure1_CloseRequestFcn(hObject, eventdata, handles)
9	delete(hObject); figure(Home);
	. . .

Pada potongan *source code* tabel 5.6, terdapat sintaks *axes*, *imread*, *set*, *delete* dan *figure*. Sintaks *axes*, *imread*, *set* telah dijelaskan pada beberapa bagian sebelum ini. Hanya saja sintaks *axes* pada potongan *source code* diatas juga diatur posisi dari *axes* tersebut diletakkan pada tampilan. Sintaks *uistack* digunakan untuk mengatur *axes* ditampilkan dibagian paling belakang sebagai latar belakang. Sedangkan *delete* dan *figure* telah dijelaskan pada gambar 5.5

5.2.2 Pengujian Black Box

Pengujian *black box* dilakukan untuk menguji apakah sistem yang dikembangkan sesuai dengan apa yang tertuang dalam spesifikasi fungsional sistem. *Black box* juga digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak yang dirancang. Kebenaran perangkat lunak yang diuji hanya dilihat berdasarkan keluaran yang dihasilkan dari data atau kondisi masukan yang diberikan untuk fungsi yang ada tanpa melihat bagaimana proses untuk mendapatkan keluaran tersebut.

5.2.2.1 Pengujian Penerapan Algoritma *fast fourier transform*

Algoritma *fast fourier transform* digunakan untuk mengubah inputan suara menjadi frekuensi. frekuensi tersebut kemudian akan disesuaikan dengan sistem sehingga didapatkan nada default. Pengujian *black box* pada algoritma ini akan ditunjukkan dengan menampilkan hasil dari tuning gitar berdasarkan inputan yang dimasukkan kedalam aplikasi. inputan yang akan diuji adalah nada yang dihasilkan dari salah satu senar pada gitar. Ini dilakukan untuk membuktikan apakah algoritma ini dapat melakukan tuning gitar.

Pada pengujian ini, inputan nada dari senar pertama berhasil dituning dengan menggunakan aplikasi ini, dengan demikian hal ini membuktikan bahwa algoritma ini cocok untuk digunakan pada tuning gitar. Gambar 5.10 menunjukkan tampilan dari hasil pengujian ini. Pada Gambar 5.10 terlihat bahwa inputan nada dari senar pertama berhasil dituning dan gambar 5.11 menampilkan proses tuning.



Gambar 5.10 Hasil Pengujian Terhadap algoritma FFT



Gambar 5.11 Hasil Pengujian Terhadap algoritma FFT

5.2.2.2 Hasil Pengujian Penerapan algoritma FFT dengan jarak 10 cm

Pengujian berikutnya adalah pengujian *black box* terhadap algoritma FFT dengan menggunakan jarak 10 cm menggunakan gitar non elektrik.

Table 5.7 Frekuensi Uji FFT dengan jarak 10 cm dalam satuan Hz

Gitar Non Elektrik Jarak 10 cm						
No	Frekuensi (Hz)					
	Senar 1	Senar 2	Senar 3	Senar 4	Senar 5	Senar 6
1	658,805	494,546	390,5278	294,0894	219,941	165,1306
2	658,602	495,584	390,7437	294,1494	219,451	165,0664
3	658,440	495,065	391,8146	294,7264	220,382	164,682
4	658,449	495,421	391,4809	294,4967	219,686	164,7561
5	658,438	496,028	390,938	293,5111	219,089	164,8153
6	658,485	494,554	391,8459	293,6078	219,604	165,0353
7	659,551	495,132	391,090	294,6143	219,326	165,2334
8	657,790	495,310	391,8341	293,9242	219,180	165,0146
9	658,569	494,547	390,9631	294,2928	219,271	164,364
10	658,293	495,494	391,6927	294,7367	219,670	164,9862
R. Frekuensi	658,542	495,168	391,293	294,215	219,560	164,908
F. Standar	659,260	493,880	392,000	293,660	220,000	164,81
Error	0,718	1,288	0,707	0,555	0,440	0,098

Pada table 5.7 terdapat R. Frekuensi, F. Standar dan Error. R. Frekuensi adalah nilai rata-rata frekuensi yang didapatkan dari 10 percobaan, sedangkan F. Standar adalah frekuensi standar yang dimiliki oleh tiap senar. Error adalah nilai selisih antara F.standar dan R. Frekuensi. Pada Tabel 5.7 terlihat bahwa error terbesar adalah 1,288 yaitu pada senar dua sedangkan error terkecil pada senar enam sebesar 0,098. Dari error tersebut dapat kita hitung nilai rata-rata error adalah 0,634 . Hal ini berarti tingkat akurasi pada jarak 10 cm adalah 99,366 %.

5.2.2.3 Hasil Pengujian Penerapan algoritma FFT dengan jarak 20 cm

Pengujian berikutnya adalah pengujian *black box* terhadap algoritma FFT dengan menggunakan jarak 20 cm menggunakan gitar non elektrik.

Table 5.8 Frekuensi Uji FFT dengan jarak 20 cm dalam satuan Hz

Gitar Non Elektrik Jarak 20 cm						
No	Frekuensi (Hz)					
	Senar 1	Senar 2	Senar 3	Senar 4	Senar 5	Senar 6
1	659,359	495,968	391,394	294,260	219,0319	164,017
2	658,729	495,738	391,3561	293,8364	219,548	164,0577
3	658,451	495,055	391,4594	294,4692	219,8036	164,6565
4	658,110	494,610	391,9342	293,0977	219,5979	165,5784
5	658,248	494,602	391,1485	295,030	219,7306	164,316
6	658,361	493,321	391,8992	294,363	220,3389	164,7335
7	658,980	495,037	391,1217	295,2254	220,3081	164,7307
8	660,982	494,838	391,6782	294,0536	219,727	164,1145
9	658,523	495,551	390,5139	294,4208	220,0864	164,0784
10	658,603	494,783	391,850	294,460	219,0126	164,0381
R. Frekuensi	658,835	494,950	391,435	294,322	219,719	164,432
F. Standar	659,260	493,880	392,000	293,660	220,000	164,81
Error	0,425	1,070	0,565	0,662	0,281	0,378

Pada table 5.8 terdapat R. Frekuensi, F. Standar dan Error. R. Frekuensi adalah nilai rata-rata frekuensi yang didapatkan dari 10 percobaan, sedangkan F. Standar adalah frekuensi standar yang dimiliki oleh tiap senar. Error adalah nilai selisih antara F. standar dan R. Frekuensi. Pada table 5.8 terlihat bahwa error terbesar dimiliki oleh senar kedua dengan error 1,070 sedangkan error terkecil pada senar lima yaitu 0,281. Dari error tersebut dapat kita hitung nilai rata-rata error adalah 0,564 dan tingkat akurasinya sebesar 99,436. Hal ini berarti, jarak 20 cm ternyata lebih baik dari pada jarak 10 cm, ini dipengaruhi oleh kuat intensitas suara senar gitar.

5.2.2.4 Hasil Pengujian Penerapan algoritma FFT dengan jarak 30 cm

Pengujian berikutnya adalah pengujian *black box* terhadap algoritma FFT dengan menggunakan jarak 30 cm menggunakan gitar non elektrik.

Table 5.9 Frekuensi Uji FFT dengan jarak 30 cm dalam satuan Hz

Gitar Non Elektrik Jarak 30 cm						
No	Frekuensi (Hz)					
	Senar 1	Senar 2	Senar 3	Senar 4	Senar 5	Senar 6
1	658,580	496,196	391,8879	294,7272	220,379	165,243
2	658,236	495,565	391,760	294,0739	218,9068	164,6475
3	658,668	496,280	391,3211	295,260	218,9739	165,0161
4	657,632	495,357	391,1494	293,9162	218,9828	165,0127
5	658,803	495,676	390,3337	294,6714	219,3267	164,978
6	659,069	495,271	391,0091	294,839	219,0837	164,4667
7	658,202	496,003	390,9025	294,393	219,3453	163,9743
8	658,451	495,956	390,8223	294,3868	219,0073	165,4987
9	657,210	495,076	390,770	294,8459	218,860	164,6648
10	658,392	496,081	391,2563	293,590	219,7243	164,9037
R. Frekuensi	658,324	495,746	391,121	294,470	219,259	164,841
F. Standar	659,260	493,880	392,000	293,660	220,000	164,81
Error	0,936	1,866	0,879	0,810	0,741	0,031

Pada table 5.9 terdapat R. Frekuensi, F. Standar dan Error. R. Frekuensi adalah nilai rata-rata frekuensi yang didapatkan dari 10 percobaan, sedangkan F. Standar adalah frekuensi standar yang dimiliki oleh tiap senar. Error adalah nilai selisih antara F. standar dan R. Frekuensi. Pada table 5.9 terlihat bahwa error terbesar dimiliki oleh senar dua dengan error 1,866 sedangkan error terkecil adalah senar enam sebesar 0,031. Dari error tersebut dapat kita hitung nilai rata-rata error adalah 0,877 dan tingkat akurasinya sebesar 99,123. Hal ini berarti, jika dibandingkan dengan hasil sebelumnya pada tabel 5.7 dan 5.8 didapatkan bahwa ternyata jarak 30 cm tidak lebih baik dibanding dengan jarak 20 cm dan jarak 10 cm, ini dapat dibuktikan dengan tabel 5.10.

5.2.2.5 Hasil Perbandingan Pengujian Algoritma FFT dengan jarak 10, 20, dan 30 cm

Dari Pengujian sebelumnya, didapatkan tabel pembandingan antara pengujian dengan jarak 10 cm, 20 cm dan 30 cm sebagai berikut :

Table 5.10 Pembanding Uji FFT dengan jarak 10, 20 dan 30 cm

Gitar Non Elektrik Jarak 30 cm						
	Frekuensi (Hz)					
	Senar 1	Senar 2	Senar 3	Senar 4	Senar 5	Senar 6
R. Error	658,324	495,746	391,121	294,470	219,259	164,841
F. Standar	659,260	493,880	392,000	293,660	220,000	164,810
Error	0,936	1,866	0,879	0,810	0,741	0,031
R. Eror	0,877					
Gitar Non Elektrik Jarak 20 cm						
	Frekuensi (Hz)					
R. Error	658,835	494,950	391,435	294,322	219,719	164,432
F. Standar	659,260	493,880	392,000	293,660	220,000	164,810
Error	0,425	1,070	0,565	0,662	0,281	0,378
R. Eror	0,564					
Gitar Non Elektrik Jarak 10 cm						
	Frekuensi (Hz)					
R. Error	658,542	495,168	391,293	294,215	219,560	164,908
F. Standar	659,260	493,880	392,000	293,660	220,000	164,810
Error	0,718	1,288	0,707	0,555	0,440	0,098
R. Eror	0,634					

Dari tabel 5.10, jarak terbaik untuk melakukan tuning gitar dengan software ini adalah 20 cm. Ini dipengaruhi oleh kuat intensitas suara senar gitar, karena pada jarak yang terlalu jauh suara akan sulit ditangkap oleh microphone. Sebaliknya jika jarak terlalu dekat, Suara yang dihasilkan senar akan menghasilkan noise.

5.2.2.6 Hasil Pengujian Algoritma FFT dengan garpu tala

Pengujian berikutnya adalah pengujian *black box* terhadap algoritma FFT dengan menggunakan garpu tala pada jarak yang berbeda.

Table 5.10 Frekuensi Uji FFT dengan garpu tala dalam satuan Hz

Nada A (220 Hz) Garpu Tala			
	Frekuensi (Hz)		
No	Jarak 30 cm	Jarak 20 cm	Jarak 10 cm
1	219,518	219,907	219,917
2	219,268	220,289	221,169
3	220,019	219,226	219,241
4	218,957	220,600	219,441

5	219,020	220,481	221,130
6	221,091	219,879	219,609
7	219,611	219,563	219,430
8	220,236	220,772	219,606
9	220,488	219,660	220,336
10	220,319	220,100	221,210
R. Error	219,853	220,048	220,109
Error	0,147	0,048	0,109

Dari tabel 5.10, jarak terbaik untuk melakukan tuning gitar dengan software ini adalah 20 cm. Hal ini sama dengan pengujian sebelumnya pada tabel 5.9, Ini dipengaruhi oleh kuat intensitas suara senar gitar, karena pada jarak yang terlalu jauh suara akan sulit ditangkap oleh microphone. Sebaliknya jika jarak terlalu dekat, Suara yang dihasilkan senar akan menghasilkan noise.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan analisa perancangan sistem, implementasi dan pengujian sistem, maka dapat disimpulkan bahwa :

1. Algoritma *Fast Fourier Transform* yang diimplementasikan pada aplikasi ini memberikan hasil yang optimal untuk tuning gitar non elektrik berdasarkan pengujian yang telah dilakukan dengan tingkat akurasi mencapai 99.436%.
2. Jarak terbaik untuk melakukan *tuning* adalah 20 cm dengan *error* mencapai 0.564. Hal ini dipengaruhi kuat intensitas suara senar gitar, karena pada jarak yang terlalu jauh. Suara akan sulit ditangkap oleh microphone, sebaliknya jika jarak terlalu dekat maka suara yang dihasilkan oleh senar akan menghasilkan noise.
3. Aplikasi ini berhasil mengimplementasikan algoritma *Fast Fourier Transform* dan mampu melakukan *tuning gitar*.

6.2 Saran

Berdasarkan analisa perancangan sistem, implementasi dan pengujian sistem, maka untuk pengembangan penelitian selanjutnya penulis menyarankan sebagai berikut:

1. Mengembangkan aplikasi *tuning* gitar ini dengan menambahkan *filter* dalam mengidentifikasi frekuensi.

2. Aplikasi ini masih menggunakan *tuning* gitar dengan *open string* (tanpa kunci) sehingga diharapkan ke depannya dapat melakukan *tuning* gitar dengan menggunakan kunci.

DAFTAR PUSTAKA

- Artikel non-personal. 8 Juli 2013 . “Gitar”, Wikipedia Bahasa Indonesia, <http://id.wikipedia.org/wiki/Gitar> , diakses 08 Januari 2014.
- Fadlisyah. 2008 . “*Pengolahan Citra Digital*”. Yogyakarta : Pt Elex Media Komputindo.
- Hanggarsari, Prativi Nugraheni, Helmy Fitriawan dan Yetti Yuniati . 2012 . “*Simulasi Sistem Pengacakan Sinyal Suara Secara Realtime Berbasis Fast Fourier Transform (FFT)*” . ELECTRICAL Jurnal Rekayasa dan Teknologi Elektro . Volume:6, No.3.
- Huda, Miftahul. 2011 . “*Konversi Nada-Nada Akustik Menjadi Chord Menggunakan Pitch Class Profile*”. Institut Teknologi Sepuluh Nopember Surabaya.
- Pradipta , Nandra. 2011. “*Implementasi Algoritma Fft (Fast Fourier Transform) Pada Digital Signal Processor (Dsp) Tms320c542*” . Program Sarjana. Universitas Diponegoro.
- Salahudin 2011. Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek). Bandung: Modula.
- Sastra, Nyoman Putra . 2008 . “*Penggunaan Algoritma Gortzel Untuk Deteksi Frekuensi Dtmf Berbasis Processor Dsp Tms320c31*”. Teknologi Elektro .Vol.3 No.2
- Selamat , Tanda dan Kevin Angkasa . 2013 . “*Perangkat Lunak Tuning Gitar dengan Menggunakan Karplus Strong Algorithm*”. Program Sarjana .Universitas STMIK IBBI Medan.
- Sugiyono. 2002. Perosesan Sinyal Digital. Yogyakarta: Andi.
- Tanudjaja , harlianto. 2007. Pengolahan Sinyal Digital & Sistem Pemrosesan Sinyal. Yogyakarta: Andi.
- Wiflihani. 2013. “*Pengetahuan Dasar Teori Musik*” . Universitas Negeri Medan.

LAMPIRAN

Lampiran A

Tabel Frekuensi Pengujian Sistem :

Lampiran A-1 Tabel Frekuensi Pengujian pada Senar Pertama

Sample Frekuensi Senar 1 (Hz)			
No	Jarak (cm)		
	10	20	30
1	658,580	658,805	659,359
2	658,236	658,602	658,729
3	658,668	658,440	658,451
4	657,632	658,449	658,110
5	658,803	658,438	658,248
6	659,069	658,485	658,361
7	658,202	659,551	658,980
8	658,451	657,790	660,982
9	657,210	658,569	658,523
10	658,392	658,293	658,603
11	658,477	658,155	659,210
12	659,212	658,713	658,651
13	658,788	658,207	660,260
14	658,848	658,003	659,004
15	658,130	658,465	657,893
16	658,111	658,519	659,548
17	658,123	658,582	659,873
18	658,499	658,344	658,625
19	657,425	657,875	658,661
20	658,217	658,340	659,196
21	658,439	660,022	659,758
22	657,657	657,738	658,951
23	658,909	658,570	657,677
24	658,643	658,332	658,960

Lampiran A-2 Tabel Frekuensi Pengujian pada Senar Kedua

Sample Frekuensi Senar 2 (Hz)			
No	Jarak (cm)		
	10	20	30
1	496,196	494,546	495,968
2	495,565	495,584	495,738
3	496,280	495,065	495,055
4	495,357	495,065	494,610
5	495,676	496,028	494,602
6	495,271	494,554	493,321
7	496,003	495,132	495,037
8	495,956	495,310	494,838
9	495,076	494,547	495,551
10	496,081	495,494	494,783
11	495,971	493,936	494,189
12	495,773	496,161	495,385
13	496,378	494,688	494,895
14	495,667	495,174	495,100
15	495,886	495,502	495,013
16	496,149	495,320	494,554
17	496,149	494,817	494,524
18	495,588	494,570	495,043
19	495,929	495,813	495,132
20	495,900	495,113	494,443
21	496,074	494,729	495,481
22	495,892	496,087	495,077
23	495,269	494,986	494,911
24	495,749	495,449	495,678

Lampiran A-3 Tabel Frekuensi Pengujian pada Senar Ketiga

Sample Frekuensi Senar 3 (Hz)			
No	Jarak (cm)		
	10	20	30
1	391,888	390,528	391,394
2	391,760	390,744	391,356
3	391,321	391,815	391,459
4	391,149	391,481	391,934
5	390,334	390,938	391,149
6	391,009	391,846	391,899
7	390,903	391,090	391,122
8	390,822	391,834	391,678
9	390,770	390,963	390,514
10	391,256	391,693	391,850
11	391,273	391,545	391,604
12	391,335	391,369	391,491
13	391,203	391,257	391,846
14	391,645	391,001	391,510
15	390,819	390,164	392,054
16	391,387	390,924	391,298
17	391,378	390,610	392,079
18	391,080	390,252	391,249
19	391,100	391,389	391,837
20	391,583	390,680	391,052
21	391,439	391,540	391,604
22	390,874	390,799	391,117
23	390,992	393,875	391,117
24	391,028	390,619	391,660

Lampiran A-4 Tabel Frekuensi Pengujian pada Senar Keempat

Sample Frekuensi Senar 4 (Hz)			
No	Jarak (cm)		
	10	20	30
1	294,727	294,089	294,260
2	294,074	294,149	293,836
3	295,260	294,726	294,469
4	293,916	294,497	293,098
5	294,671	293,511	295,030
6	294,839	293,608	294,363
7	294,393	294,614	295,225
8	294,387	293,924	294,054
9	294,846	294,293	294,421
10	293,590	294,737	294,460
11	293,782	293,844	294,456
12	293,732	293,141	293,848
13	294,313	294,010	294,521
14	294,088	294,491	293,745
15	294,652	294,724	293,866
16	294,659	293,774	293,929
17	294,032	294,099	293,583
18	293,943	293,865	295,313
19	294,804	292,878	294,090
20	294,804	294,877	293,608
21	294,543	294,146	293,789
22	294,211	294,229	294,343
23	294,334	294,179	293,370
24	294,992	293,691	293,484

Lampiran A-5 Tabel Frekuensi Pengujian pada Senar Kelima

Sample Frekuensi Senar 5 (Hz)			
No	Jarak (cm)		
	10	20	30
1	220,379	219,941	219,032
2	218,907	219,451	219,548
3	218,974	220,382	219,804
4	218,983	219,686	219,598
5	219,327	219,089	219,731
6	219,084	219,604	220,339
7	219,345	219,326	220,308
8	219,007	219,180	219,727
9	218,860	219,271	220,086
10	219,724	219,670	219,013
11	219,082	219,233	219,616
12	219,869	219,069	219,728
13	219,036	219,217	219,081
14	219,090	218,868	219,166
15	219,517	219,304	219,071
16	219,084	219,084	219,334
17	219,000	218,866	219,168
18	219,865	219,193	219,148
19	219,310	218,777	219,078
20	219,674	220,046	219,318
21	219,113	218,879	219,137
22	219,215	218,764	218,865
23	219,199	219,009	219,572
24	219,631	221,261	219,497

Lampiran A-6 Tabel Frekuensi Pengujian pada Senar Keenam

Sample Frekuensi Senar 6 (Hz)			
No	Jarak (cm)		
	10	20	30
1	165,243	165,131	164,017
2	164,648	165,066	164,058
3	165,016	164,682	164,657
4	165,013	164,756	165,578
5	164,978	164,815	164,316
6	164,467	165,035	164,734
7	163,974	165,233	164,731
8	165,499	165,015	164,115
9	164,665	164,364	164,078
10	164,904	164,986	164,038
11	164,942	165,544	164,463
12	164,123	164,514	165,357
13	165,717	164,447	164,745
14	165,040	165,535	164,925
15	164,550	164,667	164,889
16	164,905	165,012	163,917
17	164,436	164,582	165,113
18	164,264	165,176	164,600
19	164,716	165,129	165,201
20	164,624	165,729	164,713
21	164,886	164,795	164,663
22	165,453	164,794	164,585
23	165,005	165,055	164,998
24	163,865	165,345	164,559

Lampiran A-7 Tabel Frekuensi Pengujian pada Garpu Tala

Sample Frekuensi Garpu Tala 220 Hz			
No	Jarak (cm)		
	10	20	30
1	219,518	219,917	219,907
2	219,268	221,234	220,289
3	220,019	219,241	219,226
4	218,957	219,441	220,600
5	219,020	221,130	220,481
6	221,091	219,609	219,879
7	219,611	219,430	219,563
8	220,236	219,606	220,772
9	220,488	220,336	219,660
10	220,319	221,210	220,100
11	218,847	221,169	218,782
12	218,843	220,336	219,575
13	218,847	221,130	219,423

LAMPIRAN B
PENGUJIAN KODE-KODE SISTEM

Lampiran B-1 Hasil Pengujian Kode-Kode Sistem (*White Box*)

No	m.file	Kode	Keterangan Kode	Hasil Uji
1	Form Home	figure	Memanggil m.file lain	Sukses
		delete	Menghapus <i>figure</i> yang sedang ditampilkan	Sukses
		questdlg	Menampilkan kotak dialog pertanyaaan	Sukses
2	Form Tentang Gitar	figure	Memanggil m.file lain	Sukses
		delete	Menghapus <i>figure</i> yang sedang ditampilkan	Sukses
3	Form Tentang Aplikasi	figure	Memanggil m.file lain	Sukses
		delete	Menghapus <i>figure</i> yang sedang ditampilkan	Sukses
4	Form GuitarTuner	floor	Membulatkan nilai	Sukses
		mod	Menghasilkan sisa bagi	Sukses
		round	Membulatkan ke bilangan terdekat	Sukses
		semilogx	Membuat grafik pada sumbu x	Sukses
		audiorecorder	Merekam suara	Sukses
		tic	Merekam nilai pertama	Sukses
		toc	Menghasilkan waktu <i>run</i> program	Sukses
		str2double	Melakukan transformasi nilai string menjadi double	Sukses
		fft	Memanggil fasilitas algoritma <i>Fast Fourier Transform</i>	Sukses
		disp	Menampilkan nilai	Sukses
		diary	Membuat <i>file log</i>	Sukses
		datestr	Menampilkan tanggal sekarang	Sukses
		delete	Menghapus <i>figure</i> yang sedang ditampilkan	Sukses
		figure	Memanggil m.file lain	Sukses
		set	Menampilkan nilai pada <i>textfield</i>	Sukses

LAMPIRAN C
FREKUENSI SENAR

Lampiran C-1 Tabel Frekuensi Senar

Senar	Notasi Saintis	Notasi Umum	Frekuensi
Pertama	E5	e'	659,26 Hz
Kedua	B4	B	493,88 Hz
Ketiga	G4	G	392,00 Hz
Keempat	D4	D	293,00 Hz
Kelima	A3	A	220,00 Hz
Keenam	E3	E	164,81 Hz

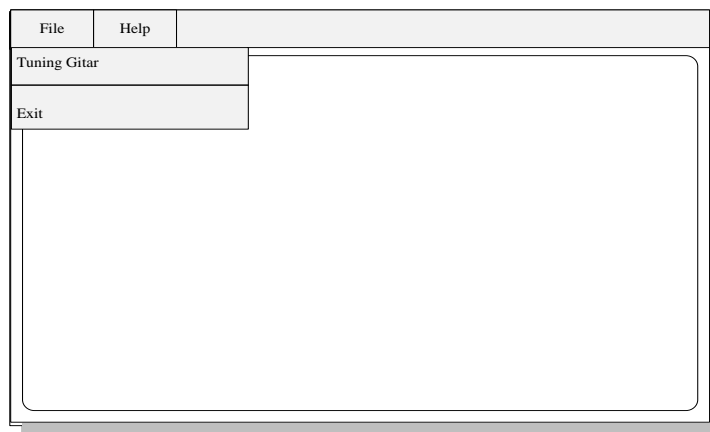
LAMPIRAN D

PERANCANGAN SISTEM

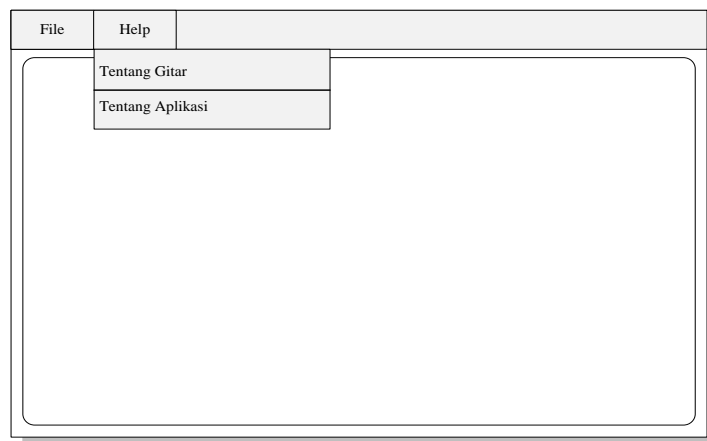
Lampiran D-1 Gambar Interface Menu Utama



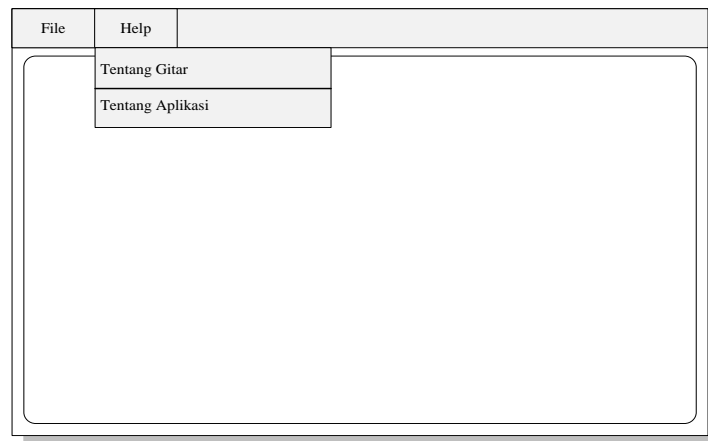
Lampiran D-2 Gambar Interface Submenu File



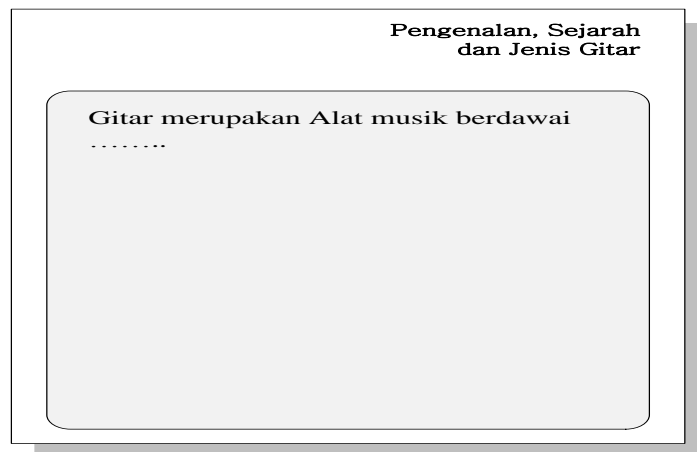
Lampiran D-3 Gambar Interface Submenu Help



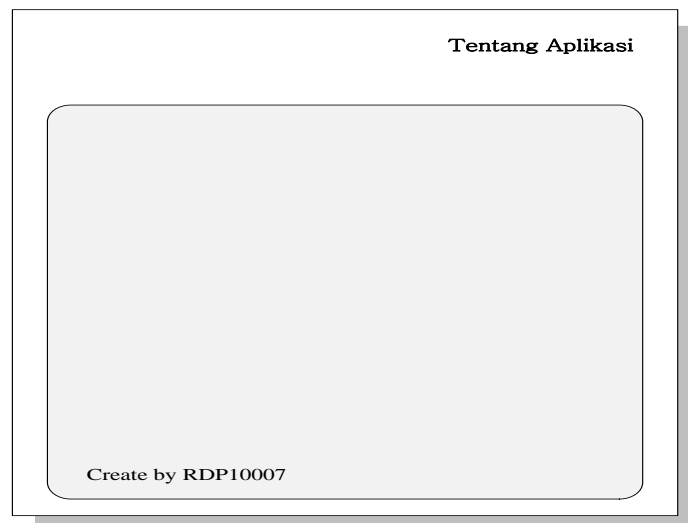
Lampiran D-4 Gambar Interface Tuning Gitar



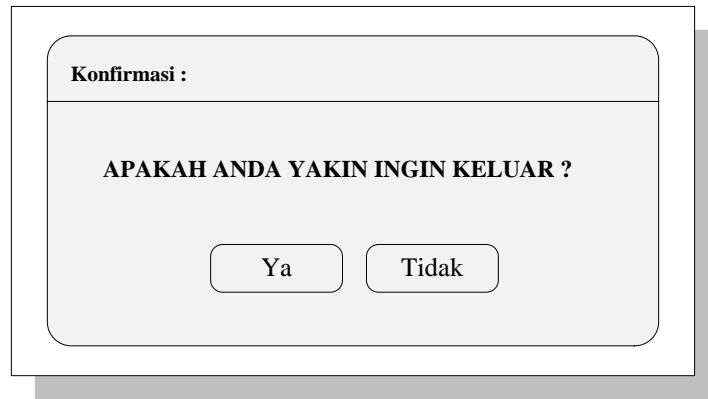
Lampiran D-5 Gambar Interface Tentang Gitar



Lampiran D-6 Gambar Interface Tentang Aplikasi



Lampiran D-7 Gambar Interface Exit



A screenshot of a software interface showing an exit confirmation dialog box. The dialog box is light gray with rounded corners and is set against a white background with a gray drop shadow. It contains a title bar at the top, a main text area with a question, and two buttons at the bottom.

Konfirmasi :

APAKAH ANDA YAKIN INGIN KELUAR ?